

Prekidi

- Prekidi
- Prekidi i instrukcijski ciklus
- Visestruki prekidi

Prekidi

- Prekidi omogućavaju da drugi moduli mogu da prekinu normalnu obradu procesora.
- Najcesce klase predida su:

Prekidi

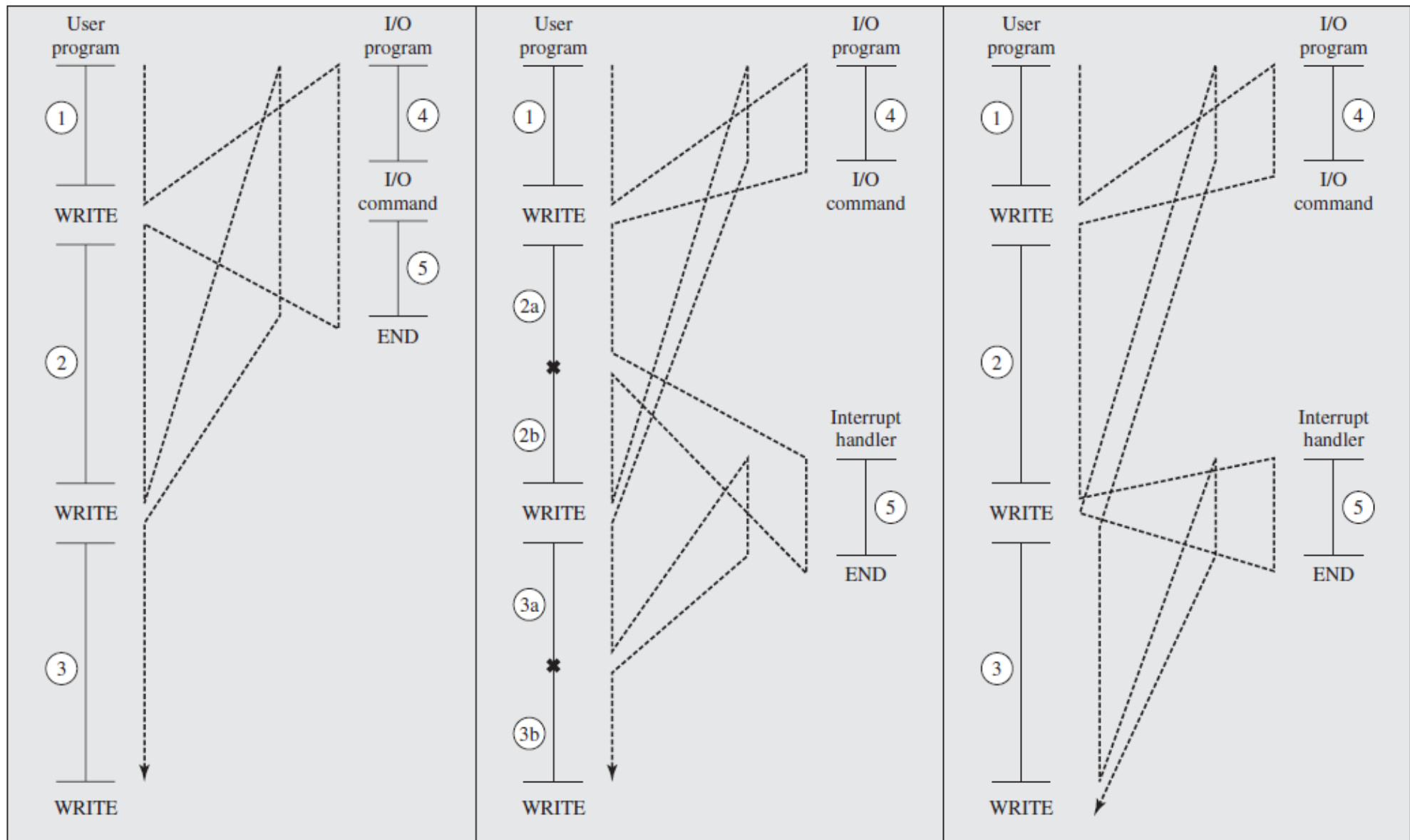
- **Programski:** Generisu ih neki uslovi koji se pojavljuju kao rezultat izvršenja instrukcije, kao na primer aritmeticko prekoracenje, deljenje nulom, pokusaj da se izvrši nekegalna masinska instrukcija, pozivanje na memoriju izvan korisnikovog dozvoljenog memorijskog prostora.

Prekidi

- **Tajmerski**: Generise ih tajmer unutar procesora.
- **UI**: Generiste ih UI kontroler da bi signalizirao dogadjaj na UI portu ili neke uslove greske.
- **Hadrverski otkaz**: Generise ih otkaz kao sto je nestanak napajanja, memorijska greska parnosti.

Prekidi I instrukcijski ciklus

- Kada spoljasnji uredjaj bude spreman da bude opsluzen, UI modul tog spoljasnjeg uredjaja salje procesoru zahtev za prekid.
- Procesor prekida operacije koje se trenutno izvrsavaju, grananjem na program za izvrsavnje prekida, koji se naziva upravljacki program signala prekida.



(a) No interrupts

(b) Interrupts; short I/O wait

(c) Interrupts; long I/O wait

Prenos upravljanja preko prekida

-
- Sa stanovista korisnickog programa, prekid je prekidanje normalne sekvence, izvršavanjem upravljackog programa prekida.
- Kada se obrada prekida završi izvršenje se nastavlja.
- Korisnicki program ne treba da sadrži nikakav specijalan kod za prihvatanje prekida, to je posao procesora i operativnog sistema.
- Da bi se prihvatili prekidi instrukcijskom ciklusu se dodaje ciklus prekida.

Prenos upravljanja preko prekida

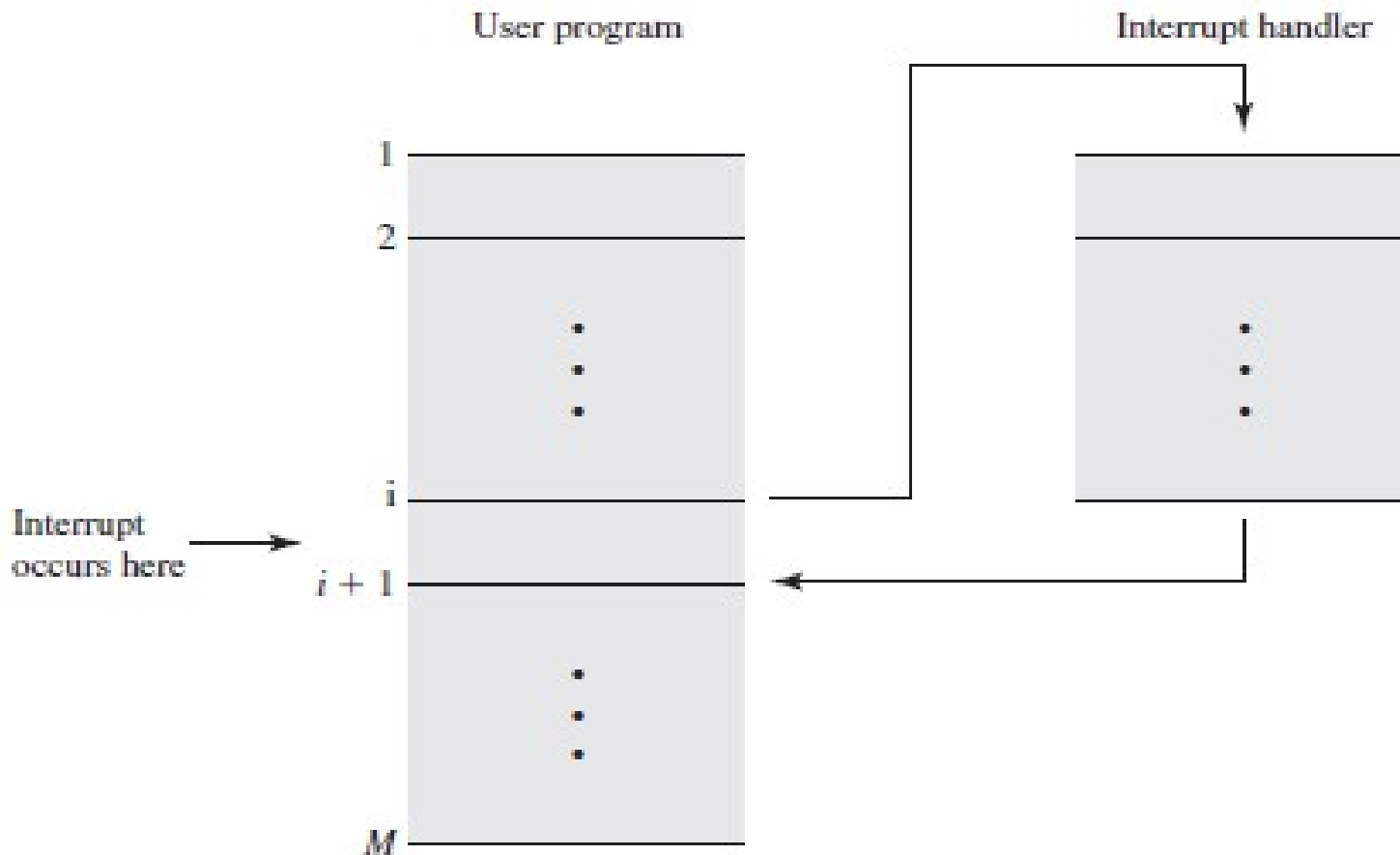


Figure 3.8 Transfer of Control via Interrupts

Prenos upravljanja preko prekida

- Da bi se prihvatili prekidi instrukcijskom ciklusu se dodaje ciklus prekida.
- U ciklusu prekida procesor proverava da li ima bilo kakvog neresenog prekida.
- Ako nema neresenog prekida, procesor prelazi u ciklus donosenja i donosi sledecu instrukciju programa koji se trenutno izvrsava.

INSTRUKCIJSKI CIKLUS SA PREKIDIMA

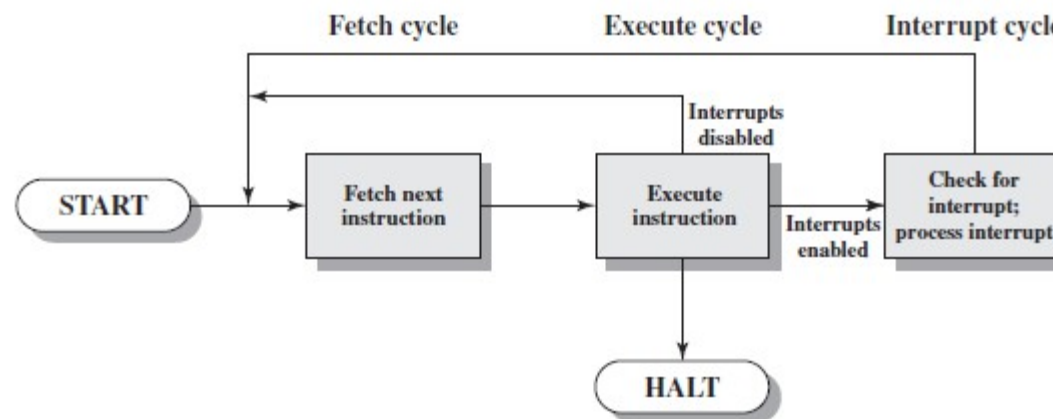


Figure 3.9 Instruction Cycle with Interrupts

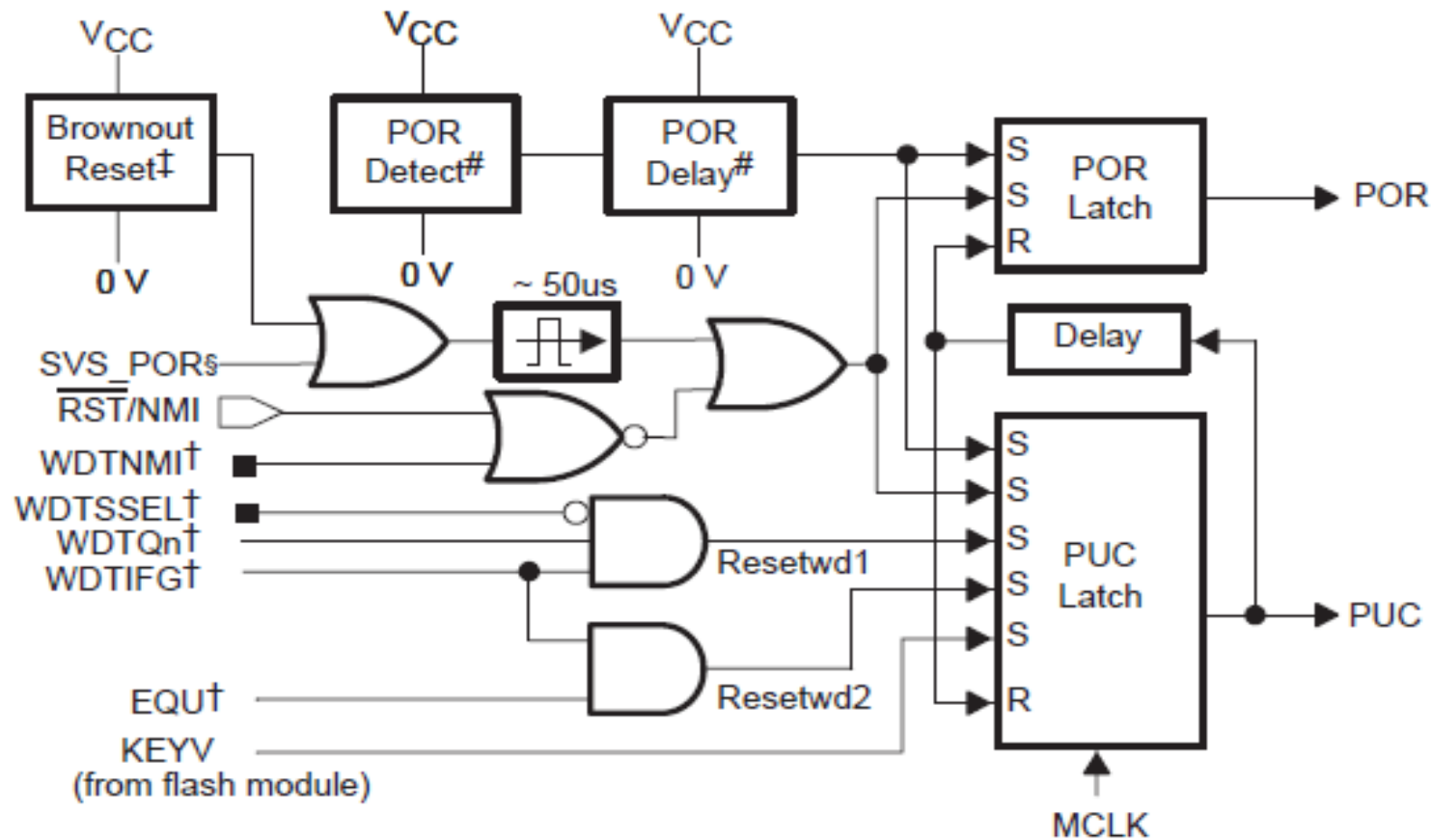
Prenos upravljanja preko prekida

- Ako postoji nereseni prekid, procesor suspenduje program koji se trenutno izvršava i čuva trenutne podatke i trenutni sadržaj programskog brojača.
- Postavlja programski broj na početnu adresu rutine upravljackog programa signala prekida.
- Nakon toga izvršava prvu instrukciju upravljackog programa prekida.
- Nakon zavretka rutine prekidnog programa, procesor može da nastavi izvršenje korisničkog programa od tačke gde je prekinut, povraćajem parametara.

Primer prekidi kod mikrokontrolera MSP430F14x

- Šema sistemskog reseta prikazana na sledecem slajdu proizvodi oba i POR (power-on reset) i PUC (power-up clear) signal.
- Različiti događaji aktiviraju signal za reset i pojavljuju se različita inicijalna stanja u zavisnosti od toga koji je signal generisan.

Primer prekidi kod mikrokontrolera MSP430F14x



† From watchdog timer peripheral module

‡ Devices with BOR only

Devices without BOR only

§ Devices with SVS only

Primer prekidi kod mikrokontrolera MSP430F14x

- POR je reset uređaja. POR generišu neki od sledeća tri događaja:
 - Uključivanje uređaja.
 - Nizak signal na RST/NMI pinu kada je konfigurisan u reset modu.
- Nisko stanje SVS (supply voltage supervisor) kada je PORON=1

Primer prekidi kod mikrokontrolera MSP430F14x

- PUC je uvek generisan kada je i POR generisan, ali ne važi obrnuto, odnosno POR ne mora biti generisan kada je PUC generisan. Neki od sledećih događaja generiše PUC:
 - POR signal
 - Watchdog timer je istekao kada je u watchdog modu
 - Prekoračenje watchdog timer bezbednosnog ključa
 - Prekoračenje bezbednosnog ključa fleš memorije

Inicijalna stanja uređaja nakon sistemskog reseta

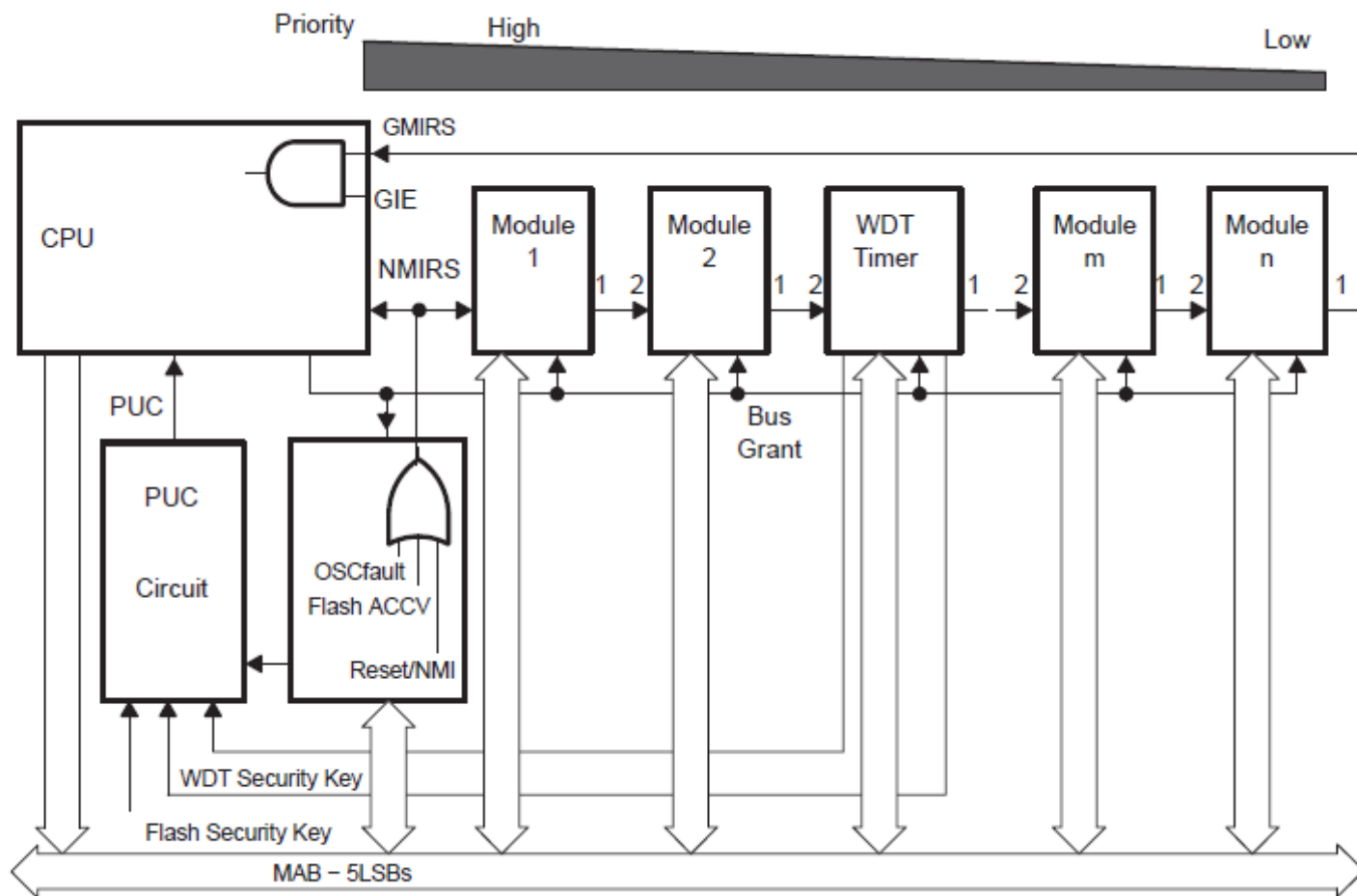
- Nakon POR inicijalna stanja MSP430 su:
 - RST/NMI pin je postavljen u reset mod.
 - I/O pinovi su prebačeni na ulazni mod.
 - Preostali periferni moduli i registri su inicijalizovani.
 - Statusni registar (SR) je resetovan.
 - Watchdog timer powers up su aktivirani u watchdog mod.
 - Programski brojč (PC) je napunjen adresom koja se nalazi u vektoru za reset (0FFFFEh). Procesor počinje sa radom od te adrese

Inicijalizacija softvera

- Nakon sistemskog reseta korisnik softvera mora da inicijalizuje procesor prema aplikacijskim potrebama, odnosno mora izvršiti sledeće:
 - Inicijalizaciju SP (stek pointer), tipično na vrh RAM-a.
 - Inicijalizaciju watchdog tajmera u skladu sa zahtevima aplikacije.
 - Konfiguraciju perifernih modula u skladu sa zahtevima aplikacije.
 - Kao dodatak, watchdog timer, oscillator fault i flash memory flags mogu da se podese kao izvori reseta.

Prioriteti prekida

- Interrupt prioriteti su fiksirani i definisani uređivanjem modula kao što je to prikazano na sledecem slajdu.
- Što je modul bliži CPU/NMIRS, to je prioritet veći.
- Interrupt prioriteti određuju koji se interrupt obrađuje kada, moguće je da u jednom trenutku, više od jednog interrupt-a čeka na izvršenje.



Vrste prekida

- Postoje tri vrste prekida:
 - System reset – sistemski reset
 - (Non)-maskable NMI – nemaskirani
 - Maskable – maskirani

(Non)-maskable Interrupts NMI – nemaskirani prekidi

- Nemaskirani prekidi ((Non)-maskable NMI interrupts) nisu maskirani glavnim bitom za dozvolu prekida – general interrupt enable bit (GIE), ali su omogućeni individualnim bitovima za dozvolu prekida (NMIE, ACCVIE, OFIE).

(Non)-maskable Interrupts NMI – nemaskirani prekidi

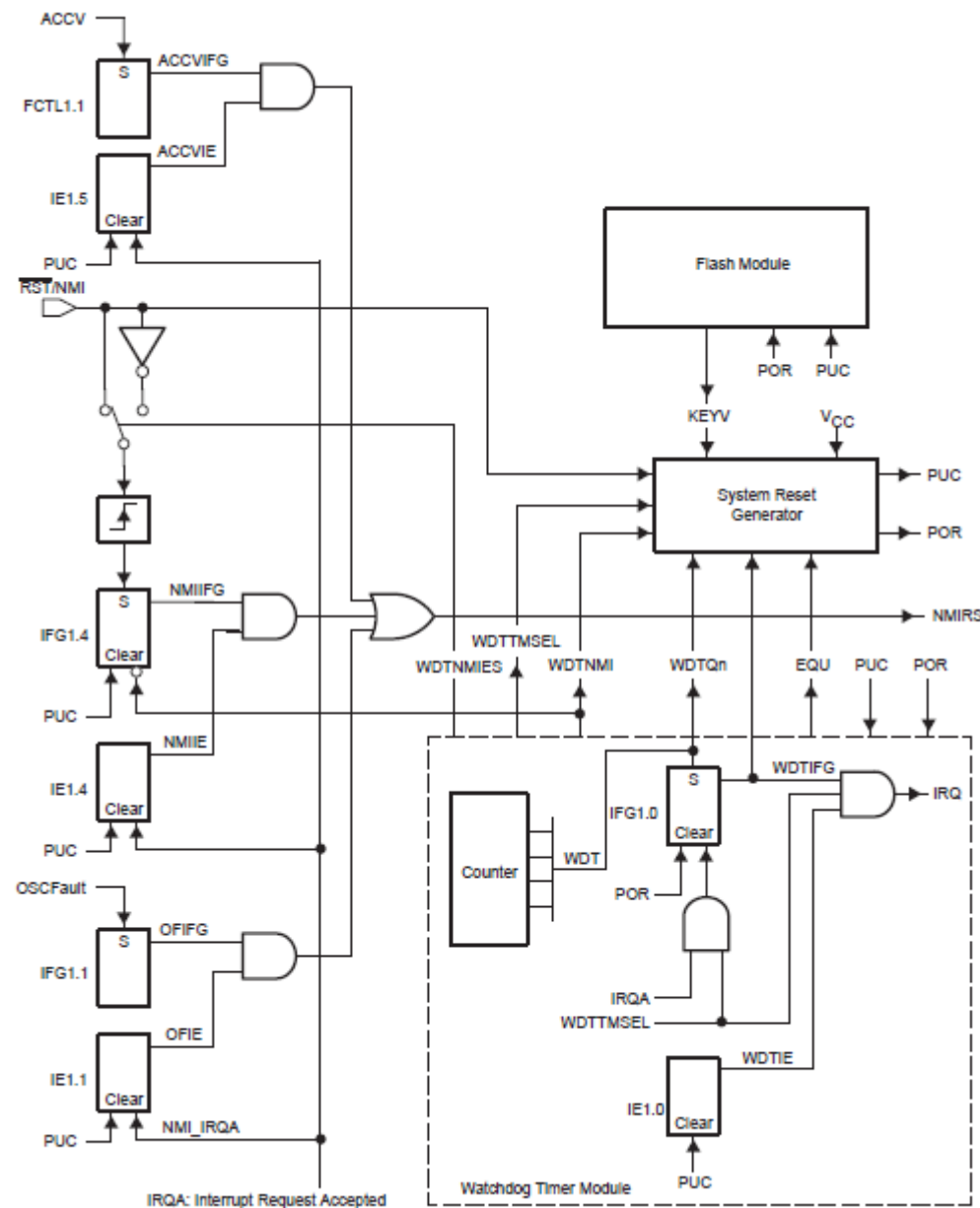
- Kada je nemaskirani prekid (NMI) prihvaćen, svi bitovi za omogućavanje NMI prekida su automatski resetovani.
- Izvršavanje programa počinje od adrese smeštene u vektoru za nemaskirane prekide((non)-maskable interrupt vector), 0FFFCh.
- Korisnički softver mora postaviti potrebne bitove za dozvolu NMI prekida tako da prekidi budu ponovo omogućeni.

Izvori nemaskiranih prekida

Izvori nemaskiranih prekida mogu biti:

- Ilica na RST/NMI pin kada je konfigurisan u NMI modu
- Dogodila se greška u radu oscilatora
- Onemogućen pristup flash memoriji

Blok sema nemaskiranih prekida



Maskable Interrupts – maskirani prekidi

- Maskirane prekide inicira periferija koja ima sposobnost da generiše prekid uključujući watchdog timer.
- Svaki izvor maskiranih prekida može biti onemogućen zasebno pomoću bita za dozvolu prekida ili svi maskirani prekidi mogu biti onemogućeni pomoću glavnog bita za dozvolu prekida (general interrupt enable bit GIE) u statusnom registru (SR).

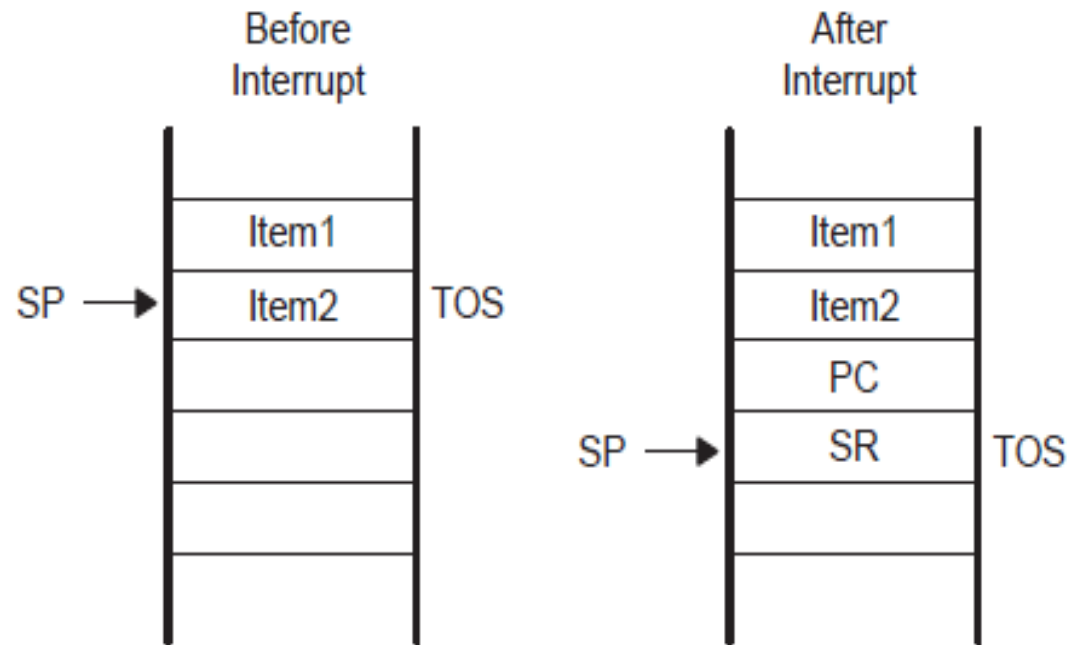
Obrada prekida

- Kada se od strane periferije zahteva prekid i kada su setovani bit za dozvolu perifernog prekida i GIE bit, zahteva se prekidna servisna rutina.
- Samo individualni bit za dozvolu mora biti setovan da bi nemaskirani prekidi mogli biti zahtevani.

Prihvatanje prekida

- Počinje sa prihvatanjem zahteva za prekid i traje dok ne počne izvršavanje prve instrukcije prekidne servisne rutine, kao što je prikazano na sledecem slajdu:

Prihvatanje prekida



Logika izvršavanja prekida

- Završi se instrukcija koja se izvršava u trenutku generisanja prekida.
- Programski brojac PC, koji pokazuje na sledeću instrukciju, smešta se na stek.
- SR (status registar) se smešta na stek.
- Ukoliko se dogodilo više zahteva za prekidom, opsluži se prekid najvišeg prioriteta.

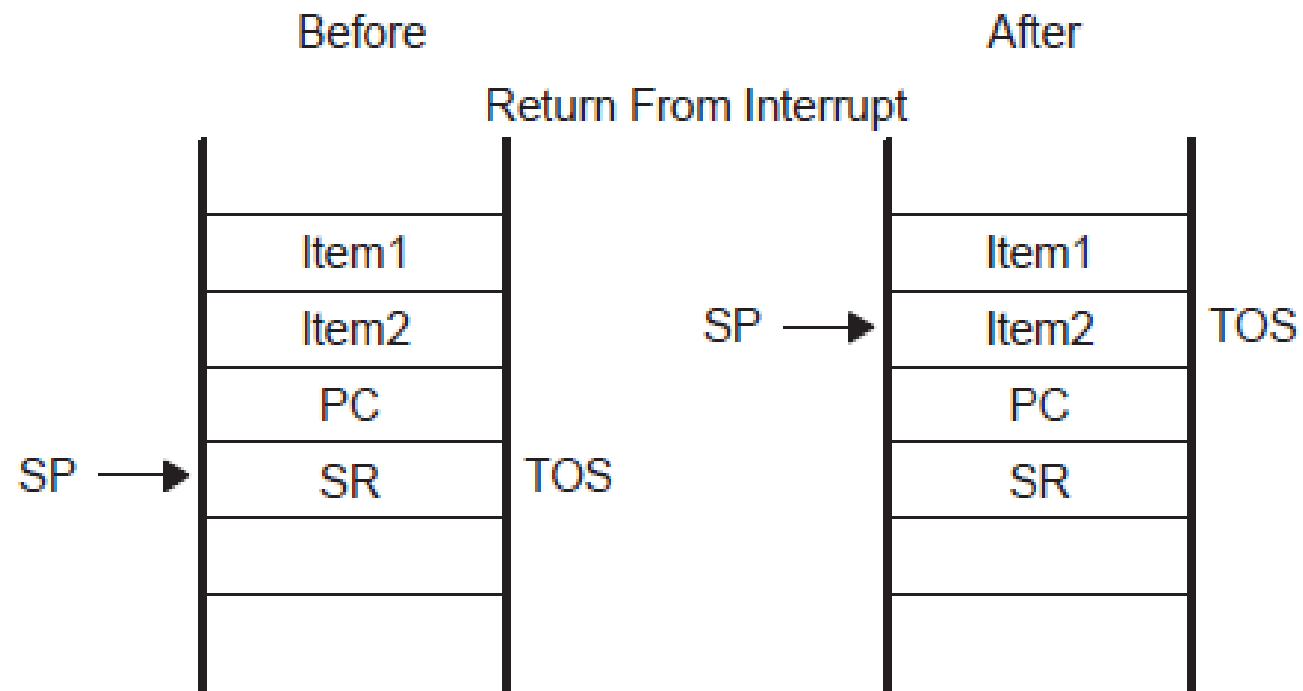
- Interrupt request flag resetuje se automatski.
- Ako je setovan flag, Multiple source flags, onda on ostaje setovan i servisira se softverski.
- SR je obrisao sa izuzetkom SCG0, koji ostaje nepromenjen.
- Zato što je GIE bit obrisao, kasniji prekidi su onemogućeni.
- Sadržaj vektora prekida se učitava u programski brojac PC:
- program nastavlja da izvršava prekidnu rutinu od date adrese.

Povratak iz prekida

RETI – povratak iz prekidne servisne rutine

Povratak iz prekida :

- Status registar SR se sa svim ranijim podešavanjima uzima sa steka.
- Sva prethodna podešavanja, uključujući GIE, CPUOFF, itd. sada su aktivirana bez obzira kako su bila postavljena tokom izvršavanja prekidne rutine.
- Programski brojač PC se uzima sa steka i program nastavlja da se izvršava od tačke gde je bio prekinut.



Vektori prekida

- Vektori prekida i adrese za startovanje su smešteni u području adresa od 0FFE0h do 0FFFEh kao što je prikazano u tabeli na sledecem slajdu.
- Vektor je programiran od strane korisnika 16-bitnim adresama analognim prekidnim servisnim rutinama

INTERRUPT SOURCE	INTERRUPT FLAG	SYSTEM INTERRUPT	WORD ADDRESS	PRIORITY
Power-up, external reset, watchdog, flash password	WDTIFG KEYV	Reset	0FFFEh	15, highest
NMI, oscillator fault, flash memory access violation	NMIIFG OFIFG ACCVIFG	(non)-maskable (non)-maskable (non)-maskable	0FFFCCh	14
device-specific			0FFFAh	13
device-specific			0FFF8h	12
device-specific			0FFF6h	11
Watchdog timer	WDTIFG	maskable	0FFF4h	10
device-specific			0FFF2h	9
device-specific			0FFF0h	8
device-specific			0FFEEh	7
device-specific			0FFECCh	6
device-specific			0FFEAh	5
device-specific			0FFE8h	4
device-specific			0FFE6h	3
device-specific			0FFE4h	2
device-specific			0FFE2h	1
device-specific			0FFE0h	0, lowest

Deklaracija prekidne rutine - Interrupt Service Routine declaration

```
// Func. declaration
Interrupt[int_vector] void myISR (Void);

Interrupt[int_vector] void myISR (Void)
{
// ISR code
}
```

```
Interrupt[TIMERA0_VECTOR] void myISR (Void);
```

```
Interrupt[TIMERA0_VECTOR] void myISR (Void)  
{  
// ISR code  
}
```